

Avalonia UI

跨平台UI开发框架

主讲：董彬

Bilibili: 兔基兔基

Github: rabbitism



Avalonia UI 是什么?

.NET Conf China
2022
开源 · 安全 · 赋能

Avalonia UI 是基于 .NET 和 XAML 的 开源跨平台 UI 框架





Avalonia UI 的特点

.NET Conf China
2022
开源 · 安全 · 赋能





Avalonia UI 的特点

可以运行于所有主流平台





Avalonia UI 的特点

可以运行于所有主流平台





Avalonia UI 的特点

可以运行于所有主流平台

开源免费 (MIT 协议)



AvaloniaUI / AvaloniaPublic

Sponsoring

Edit Pins

Unwatch446

Fork1.4k

Starred16k

<> Code

Issues1.3k

Pull requests39

Discussions

Actions

Projects4

Wiki

Security

Insights

master403 branches58 tags

Go to file

Add file

<> Code

About

maxkatz6 Merge pull request #9498 from AvaloniaUI/fixes/nc...c355aff14 hours ago21,804 commits

.github	add github sponsors	last month
.ncrunch	Ignore a few more projects.	yesterday
.nuke	Next try	2 months ago
Documentation	Update build instructions.	18 days ago
build	Enable compiled bindings by default and fix all the usages + ...	7 days ago
native/Avalonia.Native	[OSX] Use non-opaque window background for non-transpar...	4 days ago

A cross-platform UI framework for .NET

avaloniaui.net

c-sharpxamlmulti-platformgui

csharpcross-platformdotnet

dotnetcoreavaloniauser-interface

avaloniauicross-platform-xaml

Readme

MIT license

Code of conduct



Avalonia UI 的特点

可以运行于所有主流平台

开源免费 (MIT 协议)

基于 XAML 的 .NET UI 框架

与WPF的编程范式相似，全面支持MVVM

XAML 并非必须，仍可通过代码构建UI





Avalonia UI 的特点

可以运行于所有主流平台

开源免费 (MIT 协议)

基于 XAML 的 .NET UI 框架

与WPF的编程范式相似，全面支持MVVM

XAML 并非必须，仍可通过代码构建UI

像素完美的原生应用 不依赖浏览器 不依赖原生控件





Avalonia UI 的特点

可以运行于所有主流平台

开源免费 (MIT 协议)

基于 XAML 的 .NET UI 框架

与WPF的编程范式相似，全面支持MVVM

XAML 并非必须，仍可通过代码构建UI

像素完美的原生应用 不依赖浏览器 不依赖原生控件

大型项目验证的成熟框架 性能强劲





Avalonia UI 与友商的对比





Avalonia UI 与友商的对比

	Avalonia	MAUI
	Yes	Yes
	Yes	Yes
	Yes	Community
	Preview	Yes
	Preview	Yes
	Preview	No





Avalonia UI 与友商的对比

	Avalonia	MAUI
	Yes	Yes
	Yes	Yes
	Yes	Community (No)
	Preview	Yes
	Preview	Yes
	Preview	No





Avalonia UI 与友商的对比

Avalonia

MAUI

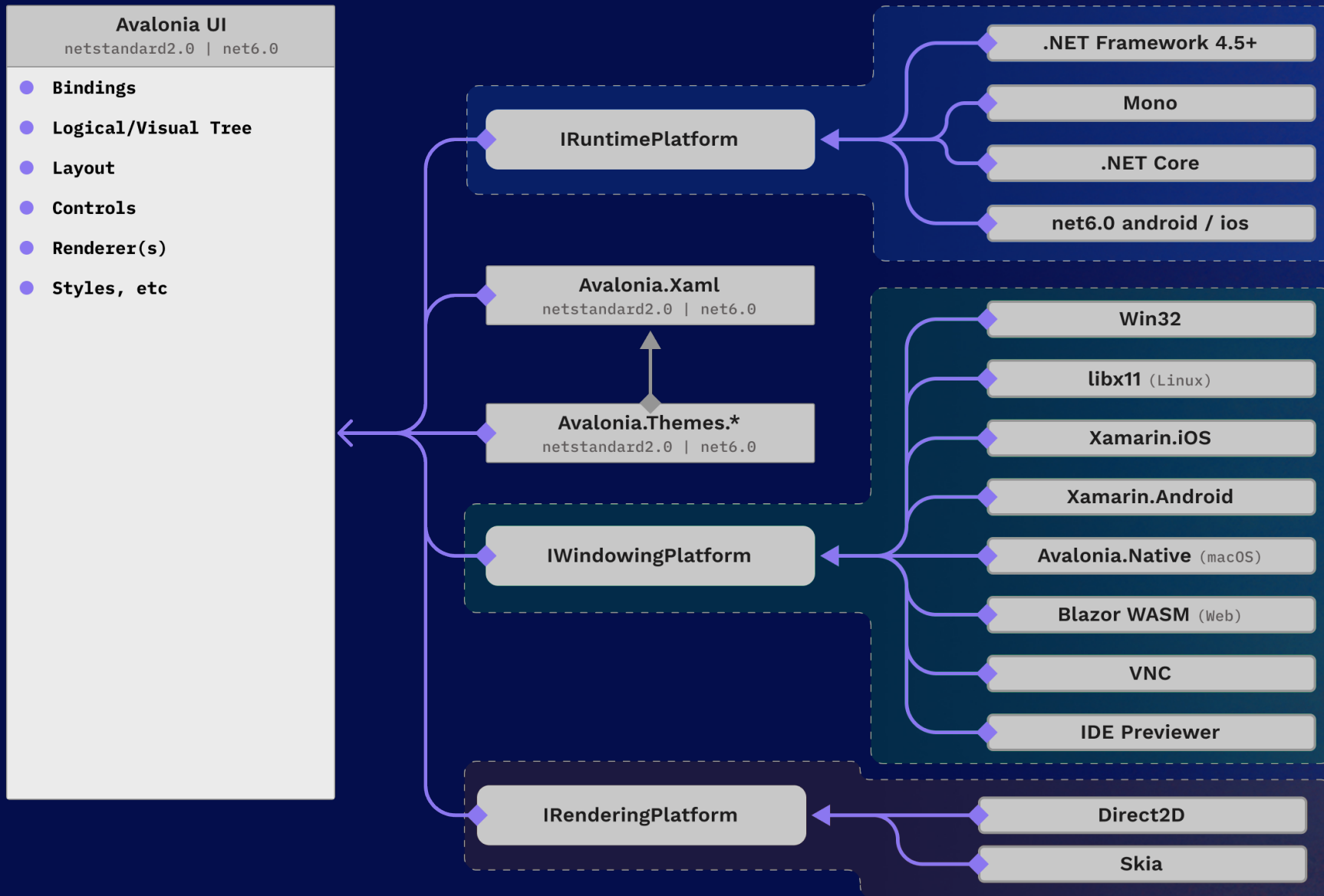
使用Skia渲染控件

原生控件的抽象层





Avalonia UI 如何实现





Avalonia UI 如何实现

Avalonia UI

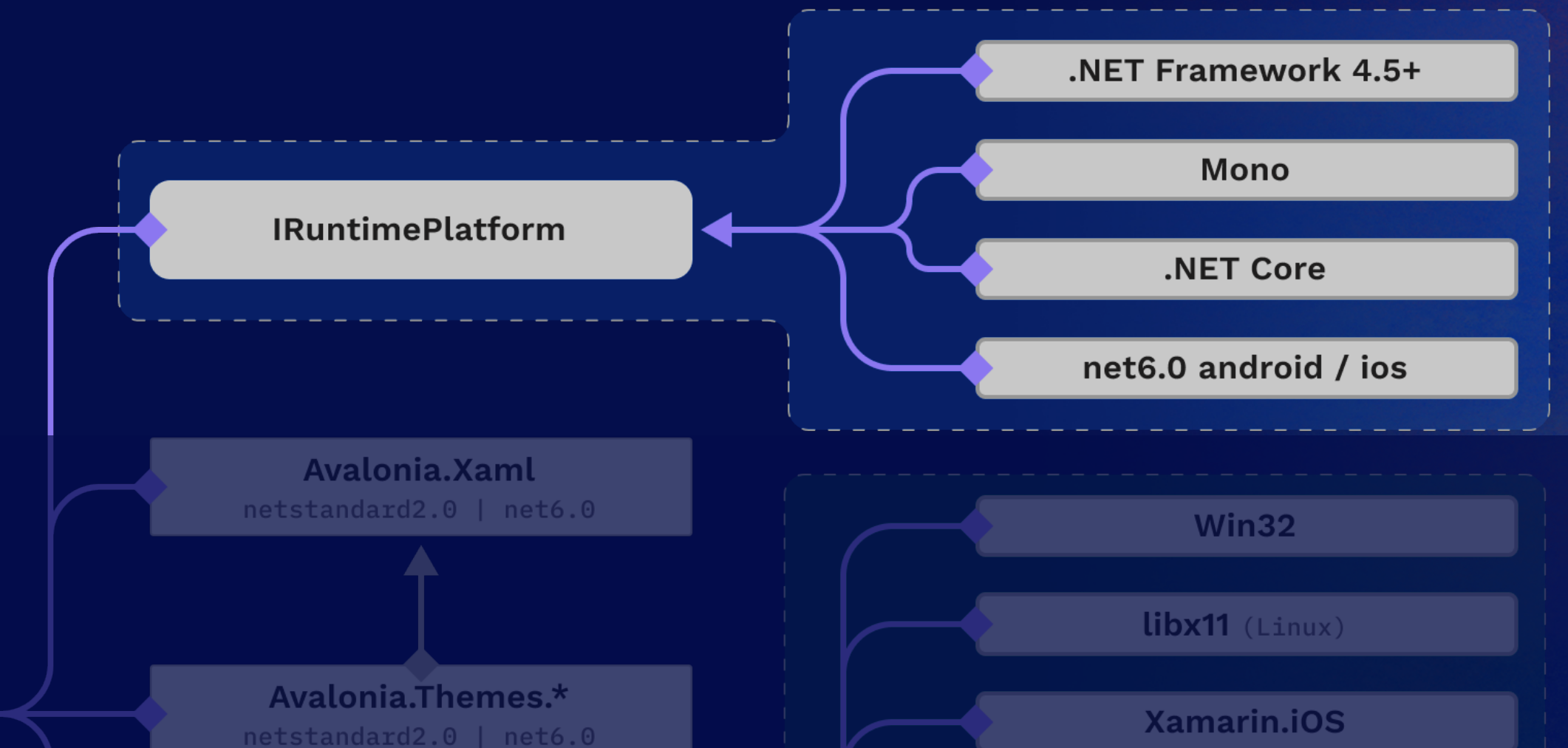
netstandard2.0 | net6.0

- Bindings
- Logical/Visual Tree
- Layout
- Controls
- Renderer(s)
- Styles, etc





Avalonia UI 如何实现





Avalonia UI 如何实现

.NET Core

net6.0 android / ios

.NET Conf China
2022
开源 · 安全 · 赋能

Avalonia.Xaml

netstandard2.0 | net6.0

Avalonia.Themes.*

netstandard2.0 | net6.0

IWindowingPlatform

Win32

libx11 (Linux)

Xamarin.iOS

Xamarin.Android

Avalonia.Native (macOS)

Blazor WASM (Web)

VNC

IDE Previewer





Avalonia UI 如何实现

Avalonia.Themes.*
netstandard2.0 | net6.0

IWindowingPlatform

IRenderingPlatform

libx11 (Linux)

Xamarin.iOS

Xamarin.Android

Avalonia.Native (macOS)

Blazor WASM (Web)

VNC

IDE Previewer

Direct2D

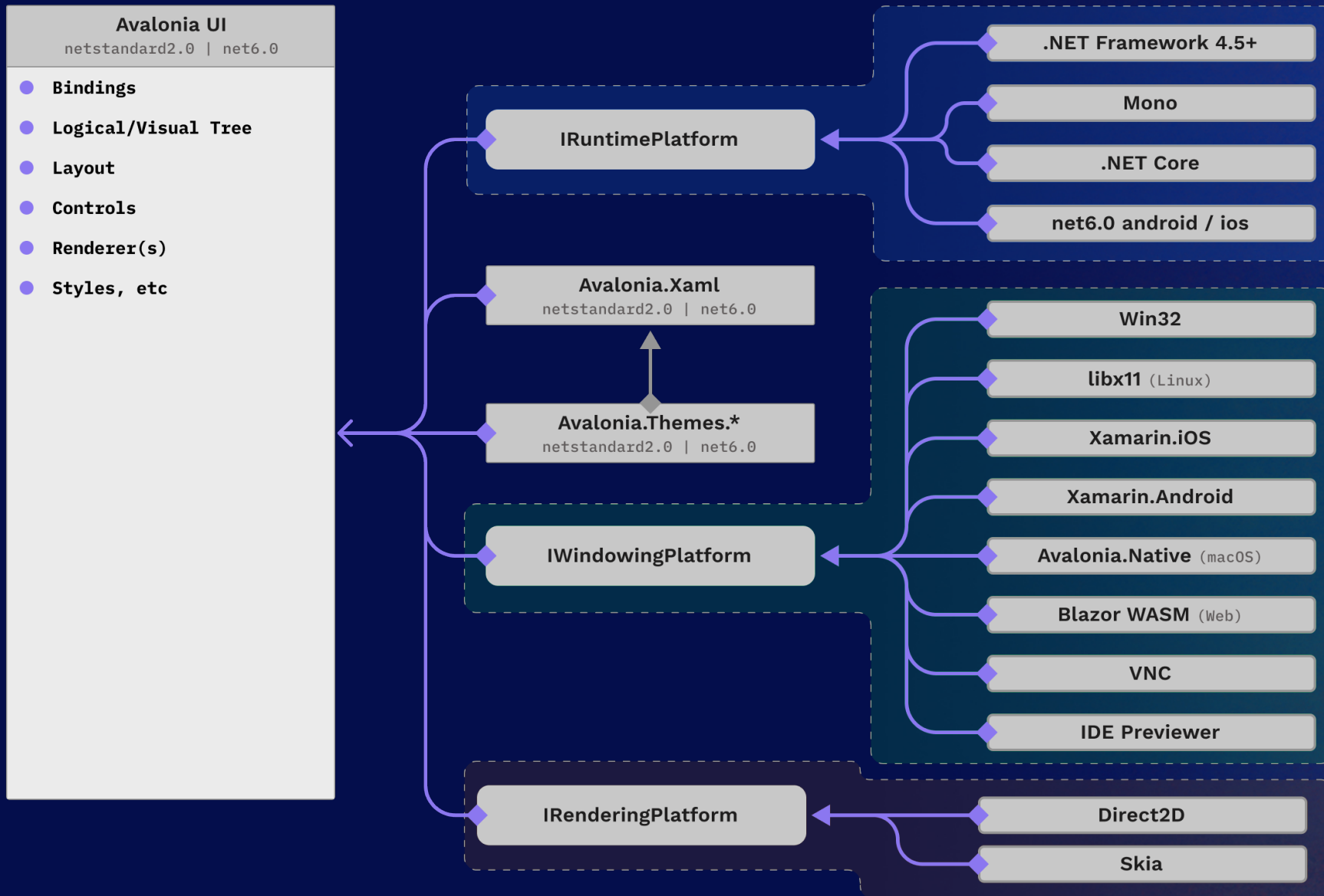
Skia

.NET Conf China
2022
开源 · 安全 · 赋能





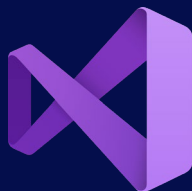
Avalonia UI 如何实现





Avalonia UI 如何开发

.NET Conf China
2022
开源 · 安全 · 赋能





Avalonia UI 如何开发

.NET Conf China
2022
开源 · 安全 · 赋能



Avalonia for Visual Studio



AvaloniaRider Plugin

XAML代码智能提示

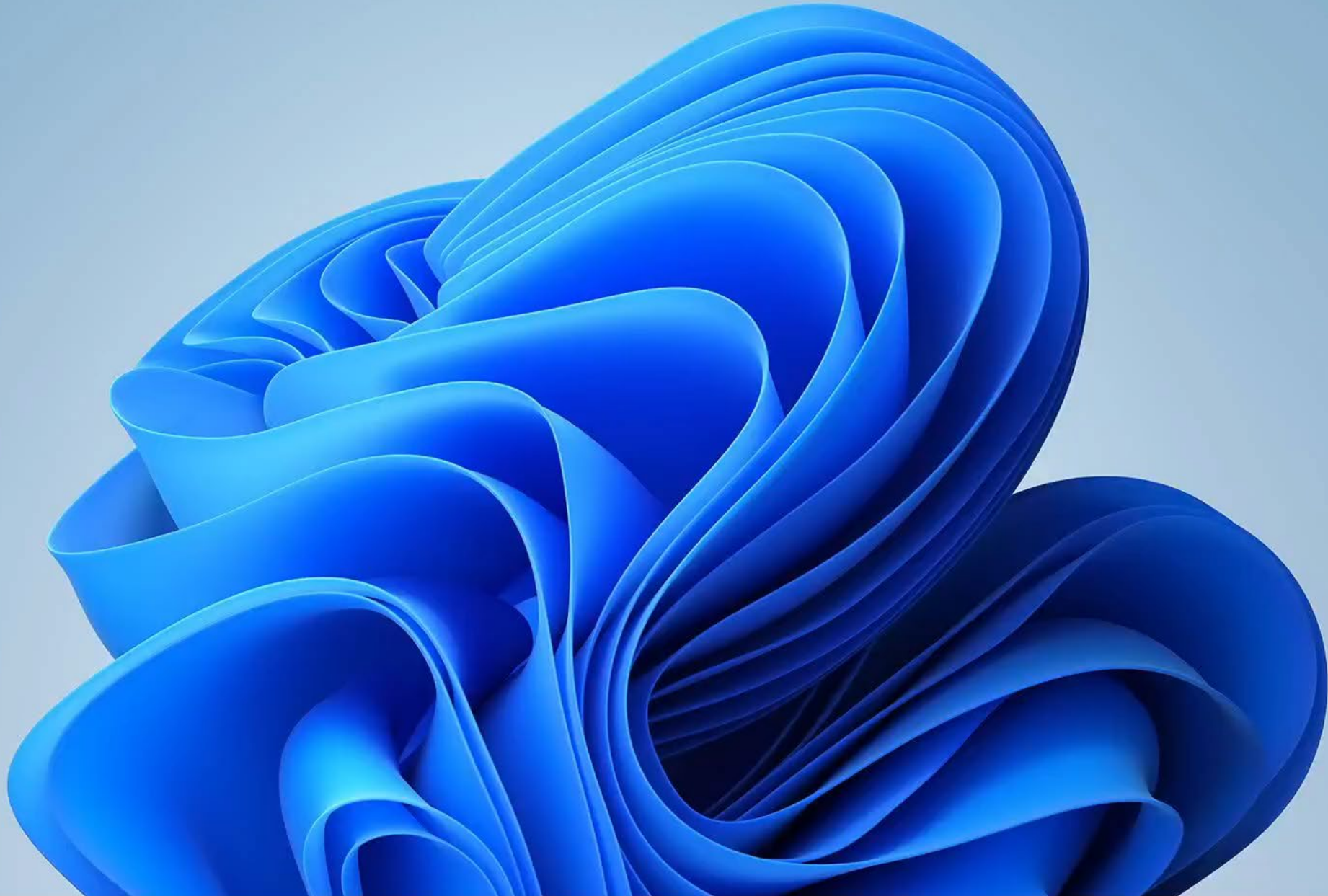
XAML语法高亮

XAML预览

默认项目/文件模板







Search



11:26 PM
11/28/2022



Avalonia UI 与 WPF 的关系

```
using Avalonia.Controls;

namespace Avalonia.NETCoreMVVMApp1.Views
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }
    }
}
```

```
<Window
    x:Class="Avalonia.NETCoreMVVMApp1.Views.MainWindow"
    xmlns="https://github.com/avaloniaui"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:vm="using:Avalonia.NETCoreMVVMApp1.ViewModels"
    Title="Avalonia.NETCoreMVVMApp1"
    d:DesignHeight="450"
    d:DesignWidth="800"
    Icon="/Assets/avalonia-logo.ico"
    mc:Ignorable="d">

    <Button
        HorizontalAlignment="Center"
        VerticalAlignment="Center"
        Content="{Binding Greeting}" />

</Window>
```





Avalonia UI 与 WPF 的关系

Avalonia 由 WPF 启发而来

编程范式相同

代码不保证兼容

对 WPF 有一定改进

Avalonia UI

netstandard2.0 | net6.0

- Bindings
- Logical/Visual Tree
- Layout
- Controls
- Renderer(s)
- Styles, etc





Avalonia UI 对 WPF 的改进

- 依赖属性

WPF

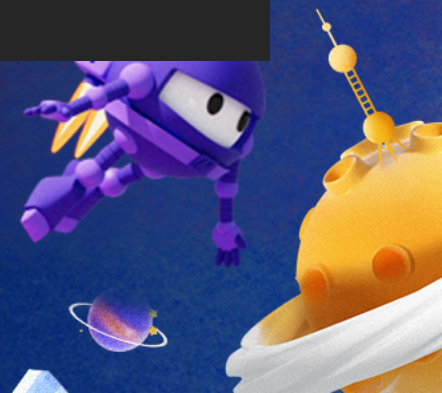
DependencyProperty

```
public int Count
{
    get { return (int)GetValue(CountProperty); }
    set { SetValue(CountProperty, value); }
}
public static readonly DependencyProperty CountProperty =
    DependencyProperty.Register(nameof(Count), typeof(int), typeof(MainWindow), new PropertyMetadata(0));
```

Avalonia

StyledProperty DirectProperty

```
public int Count
{
    get => GetValue(CountProperty);
    set => SetValue(CountProperty, value);
}
public static readonly StyledProperty<int> CountProperty =
    AvaloniaProperty.Register<MainWindow, int>(nameof(Count), 0);
```





Avalonia UI 对 WPF 的改进

- 数据模板

WPF

在MyControl.Resources中储存

分别定义 DataTemplate 和 DataTemplateSelector

Avalonia

在MyControl.DataTemplates中储存

IDataTemplate同时兼顾两者的功能





Avalonia UI 对 WPF 的改进

- 绑定

WPF

Binding完全通过反射实现

Avalonia

Binding

CompiledBinding: 编译期检查绑定是否合法

使用 x:DataType 和 x:CompileBindings 标记全局开启
使用 ReflectionBinding 在其中局部进行反射绑定





Avalonia UI 对 WPF 的改进

- 绑定至指定Name的控件

WPF

使用 ElementName

```
<TextBox Name="other" />  
<TextBlock Text="{Binding ElementName=other, Path=Text}" />
```

Avalonia

更加简洁的语法

```
<TextBox Name="other" />  
<TextBlock Text="{Binding #other.Text}" />
```





Avalonia UI 对 WPF 的改进

- 绑定至父级

WPF

使用 RelativeResource 标记拓展

```
<Border Tag="Hello World!">
    <TextBlock Text="{Binding Path=Tag, RelativeSource={RelativeSource
        Mode=FindAncestor, AncestorType=Border}}" />
</Border>
```

Avalonia

更加简洁的语法*

```
<Border Tag="Hello World!">
    <TextBlock Text="{Binding $parent[Border].Tag}" />
</Border>
```

*两者在功能上有所不同。简洁写法指逻辑树，而完整写法指视觉树





Avalonia UI 对 WPF 的改进

- 绑定至父级

WPF

使用 RelativeResource 标记拓展

```
<Border Tag="Hello World!">
    <TextBlock Text="{Binding Path=Tag, RelativeSource={RelativeSource
        Mode=FindAncestor, AncestorType=Border, AncestorLevel=1}}" />
</Border>
```

Avalonia

更加简洁的语法*

```
<Border Tag="Hello World!">
    <TextBlock Text="{Binding $parent[Border;1].Tag}" />
</Border>
```

*两者在功能上有所不同。简洁写法指逻辑树，而完整写法指视觉树





Avalonia UI 对 WPF 的改进

- 样式

WPF

Style 存储在 MyControl.Resources 中

需要在控件中指定 Style={...}

Avalonia

Style 存储在 MyControl.Styles 中

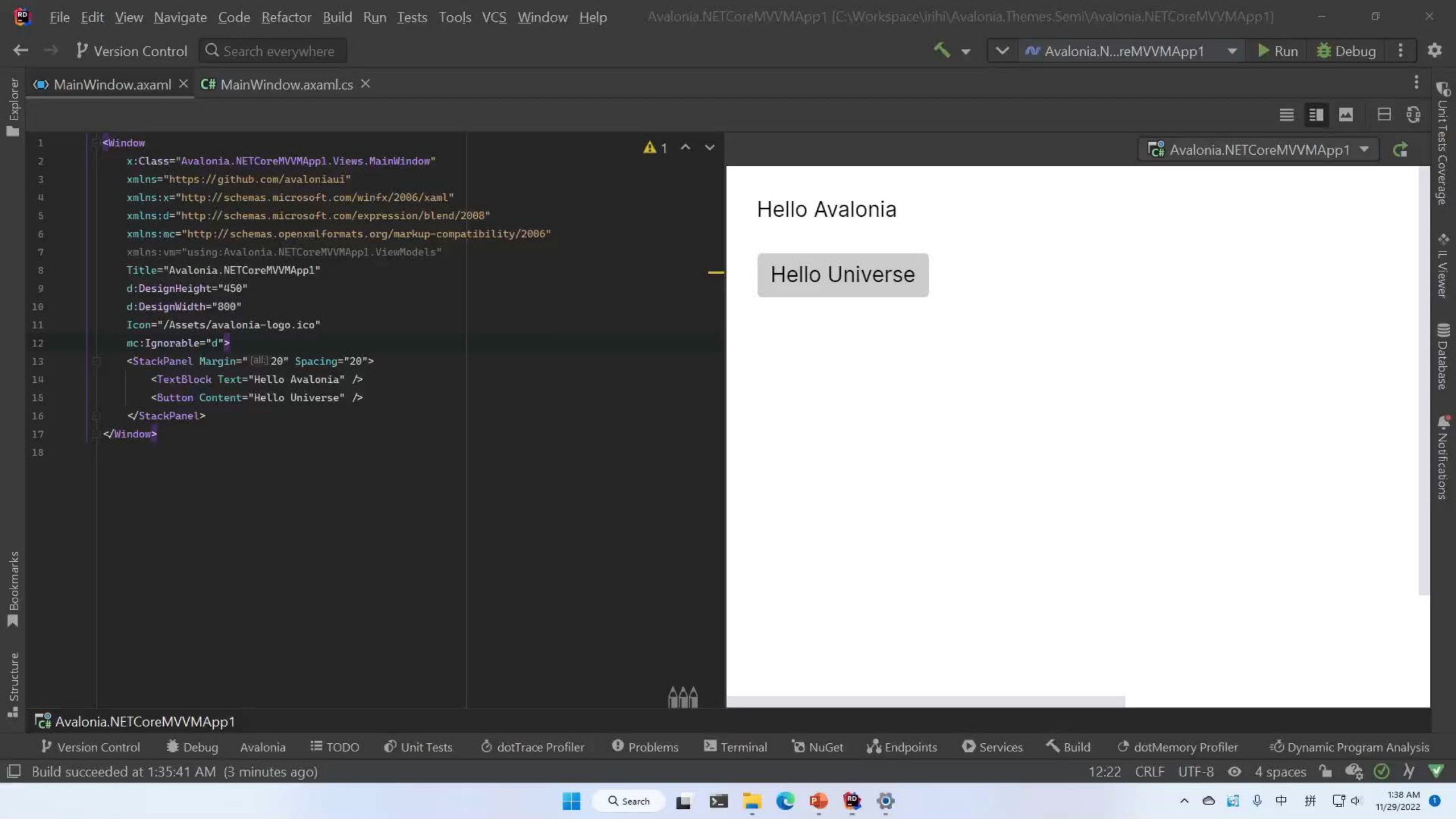
Style 通过 Selector 决定是否应用在控件上

控件拥有 Classes 属性，用法与 CSS 类似

控件内部维护 PseudoClasses，用于替代 WPF 的 VisualState

*两者在功能上有所不同。简洁写法指逻辑树，而完整写法指视觉树







Avalonia UI 对 WPF 的改进

- 样式

```
<!-- Type -->
<Style Selector="Button" />

<!-- Name -->
<Style Selector="Button#myButton" />

<!-- Class/PseudoClasses -->
<Style Selector="Button.large:focus" />

<!-- Property Equal -->
<Style Selector="Button[IsDefault=true]" />

<!-- Template Part -->
<Style Selector="Button /template/ ContentPresenter#PART_ContentPresenter" />

<!-- Not -->
<Style Selector="TextBlock:not(.h1)" />

<!-- Nth Child -->
<Style Selector="TextBlock:nth-child(2n+3)" />
```




Avalonia UI 对 WPF 的改进

- 其他

Command 直接绑定至方法

Grid.RowDefinitions = “* * * Auto”

IsVisible 取代 Visibility

{Binding !IsVisible}

...





Avalonia UI 开发的产品

.NET Conf China
2022
开源 · 安全 · 赋能





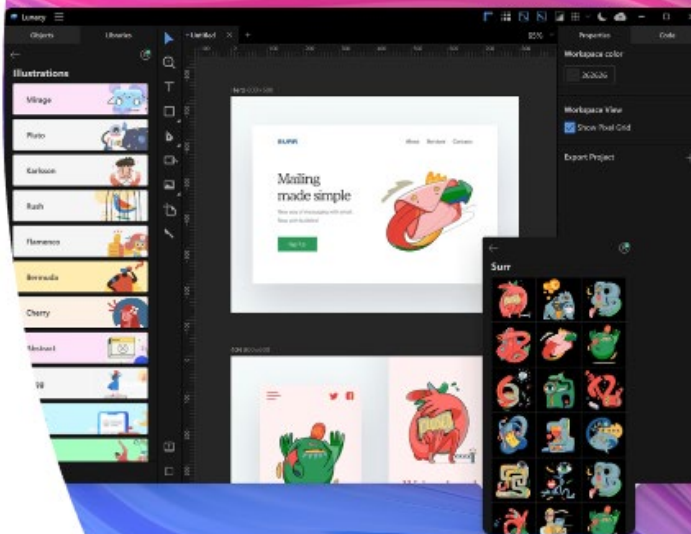
Avalonia UI 开发的产品

.NET Conf China
2022
开源 · 安全 · 赋能



Icons8 Lunacy

Free design software that keeps your flow with AI tools and built-in graphics.



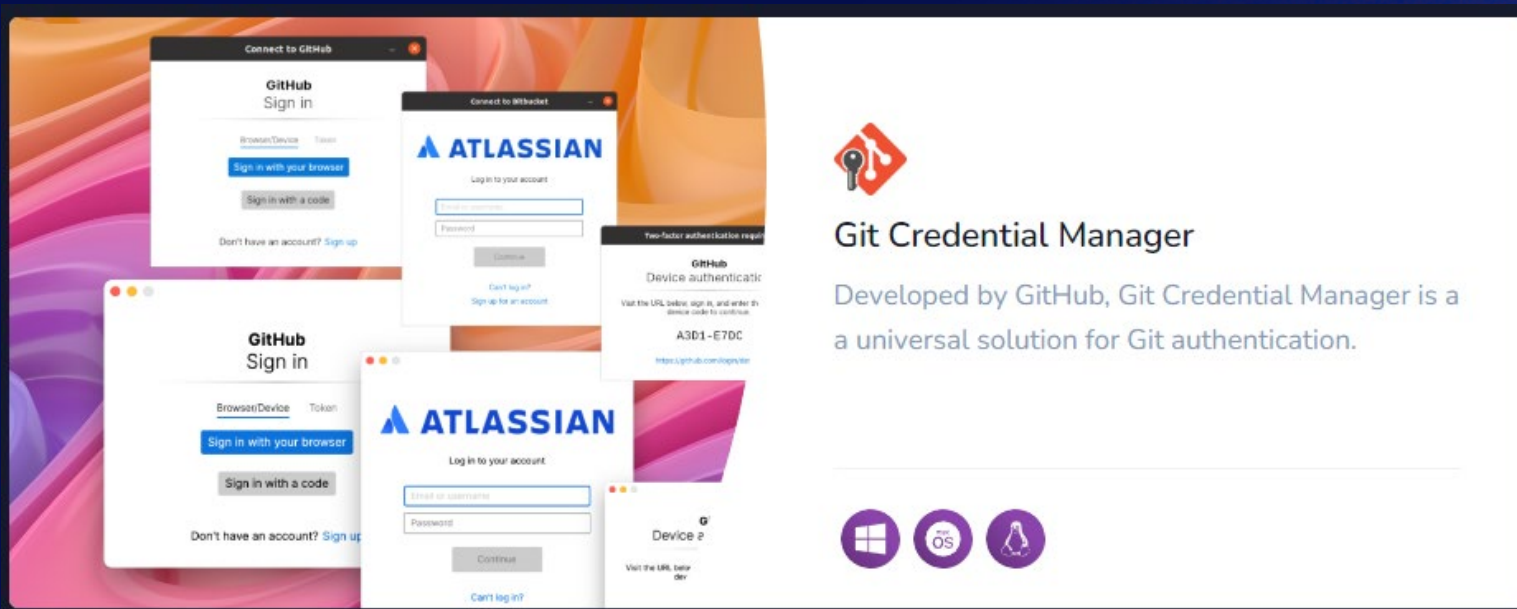
<https://icons8.com/lunacy>





Avalonia UI 开发的产品

.NET Conf China
2022
开源 · 安全 · 赋能



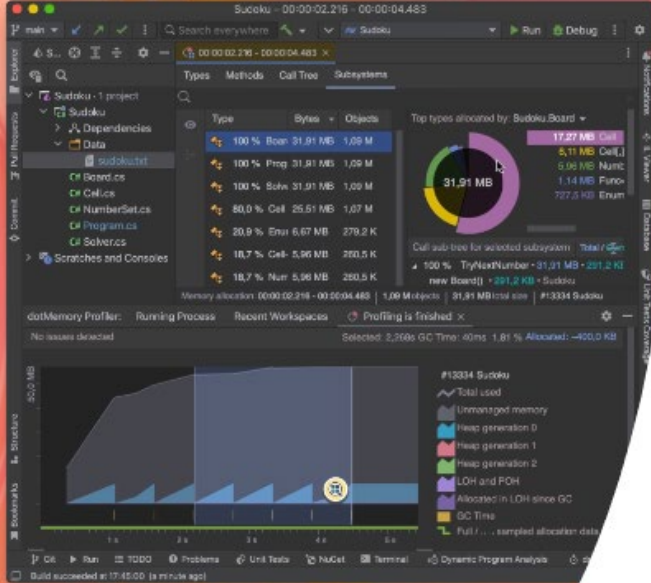
<https://github.com/GitCredentialManager/git-credential-manager>





Avalonia UI 开发的产品

.NET Conf China
2022
开源 · 安全 · 赋能



Type	Bytes	Objects
100 % Board	31.91 MB	1,09 M
100 % Prog	31.91 MB	1,09 M
100 % Solve	31.91 MB	1,09 M
80.0 % Cell	25.51 MB	1,07 M
20.9 % Enum	6.67 MB	279.2 K
18.7 % Cell	5.96 MB	260.5 K
18.7 % Num	5.96 MB	260.5 K

dotMemory Profiler: Running Process Recent Workspaces Profiling is finished

Memory allocation: 00:00:02.216 - 00:00:04.483 | 1,09 M Objects | 31.91 MB total size | #13334 Sudoku

Selected: 2,356s GC Time: 40ms 1.81 % Allocated: ~400.0 KB

#13334 Sudoku

- Total used
- Unmanaged memory
- Heap generation 0
- Heap generation 1
- Heap generation 2
- LOH and POH
- Allocated in LOH since GC
- GC Time

Full... sampled allocation data

Build succeeded at 17:45:00 (a minute ago)

JetBrains Rider

JetBrains uses Avalonia UI to enable DotMemory and DotTrace to run in Rider, their cross-platform IDE.

Windows macOS Linux

<https://www.jetbrains.com/profiler/>





Avalonia UI 开发的产品

.NET Conf China
2022
开源 · 安全 · 赋能



<https://mark-10.com/products/motorized-force-testers/series-f-easymesur/>





开源 · 安全 · 赋能

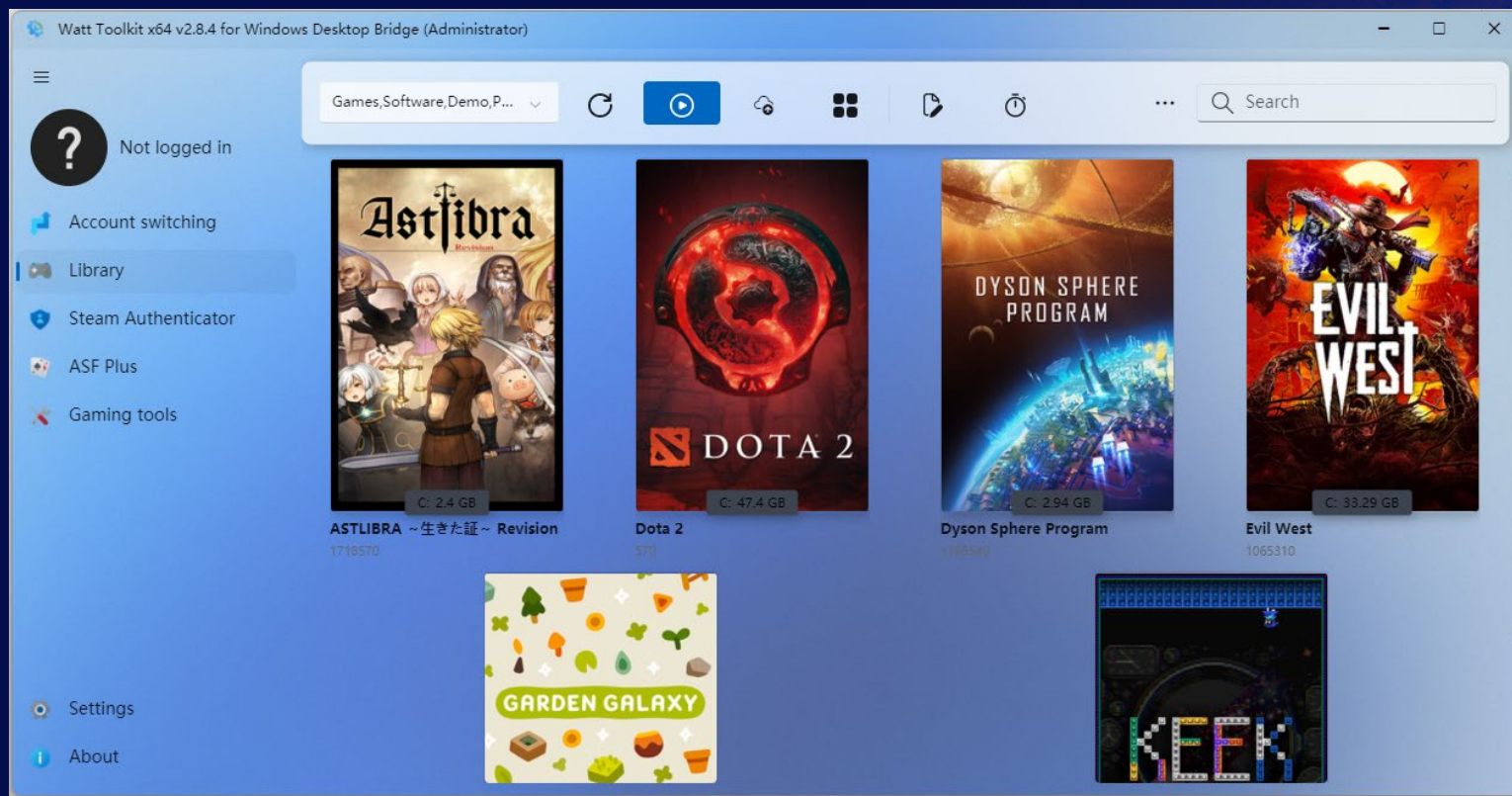


<https://vhdplus.com/>



Avalonia UI 开发的产品

.NET Conf China
2022
开源 · 安全 · 赋能



<https://steampp.net/>





Avalonia UI V11-preview

.NET Conf China
2022
开源 · 安全 · 赋能



- Compositional 渲染
- Wayland 支持
- Control Theme
- NativeAOT
- Accessibility and Automation
- Performance
- 输入法
- CJK/Unicode/Emoji
- 富文本渲染排版
- 3D 转换





Avalonia UI V11

.NET Conf China
2022
开源 · 安全 · 赋能

Avalonia

MAUI

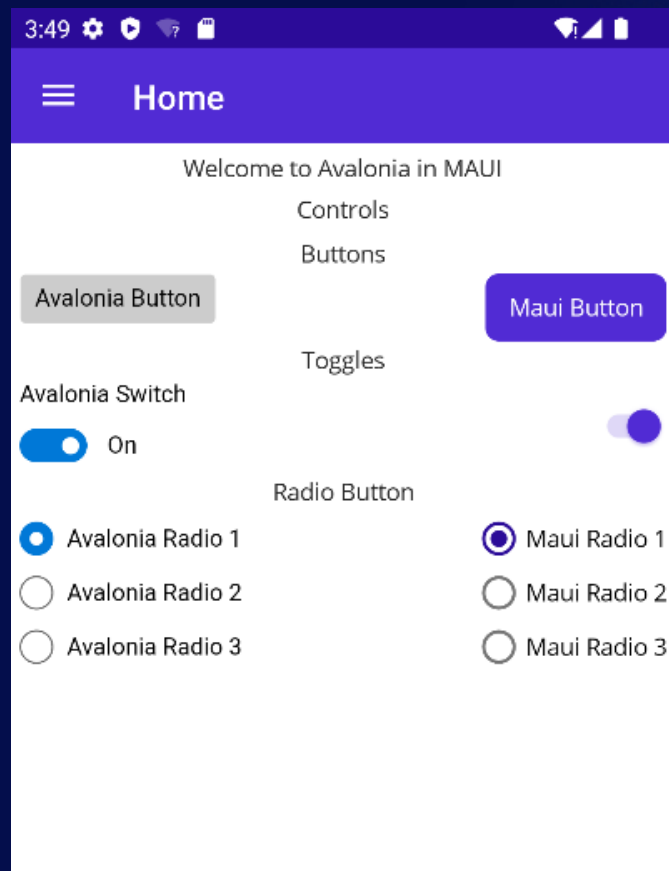




Avalonia UI V11

.NET Conf China
2022
开源 · 安全 · 赋能

Avalonia MAUI



<https://github.com/AvaloniaUI/AvaloniaMauiHybrid>



官方网站

<https://avaloniaui.net>

GitHub 仓库

<https://github.com/AvaloniaUI/Avalonia>

中文文档

<https://github.com/avaloniachina/Documentation>



Avalonia 中文社区





.NET Conf China
2022
开源 · 安全 · 赋能

THANKS!

感谢聆听